Inferring 3D shape from incomplete 2D pose using a Gaussian prior

Dan B Goldman[1], Nate Reid[2], and Doug Epps[2]
[1]Adobe, [2]ImageMovers Digital

2008*

Abstract

In this paper we construct a simple full-covariance Gaussian prior for person-specific facial pose at a number of points on the face. When the relative orientation and translation of the camera and head are known, this simple model permits triangulation and correspondence to be expressed in a straightforward manner: For orthographic cameras, triangulating points amounts to linear regression, and for perspective cameras, we iteratively approach the solution by linearizing about the current pose. The correspondence problem under this prior is NP-hard, but we have achieved good correspondence by initializing using linear assignment, then searching locally using pairwise swaps in the correspondence matrix. Using facial motion capture data, we show that previous models of facial pose relying on subspace constraints can produce significant errors unless high numbers of dimensions are used, and evaluate the performance of our prior for both the triangulation and correspondence problems.

1 Introduction

Facial motion capture is used in the film and games industries to acquire actors' performances for later retargeting to digital characters. A variety of techniques have been developed to acquire facial motion. Active sensing methods [1] have acquired relatively high-resolution facial shape, but such systems are unsuitable for performance capture due to the small performance space and use of projectors shining directly into the performers' faces. Infrared optical systems like those from Vicon [2] use many cameras with motion capture markers (reflective spheres) affixed to actors faces. Such systems can obtain high-quality results, but these markers can physically interfere with an actor's performance. Many recent works have attempted to track facial motion using unobtrusive "makeup" markers [3], or no markers at all [4, 5]. However, most such work relies on low-dimensional subspace models and monocular weak perspective camera models, making them less appropriate for high precision data capture required by the entertainment industry.

*This paper was written in 2008 but subsequently unpublished until 2012. References may be out of date.

In this paper we demonstrate a straightforward method to obtain high-precision facial motion capture from 2D tracks in multiple perspective views, with missing or uncertain data, using a 3D prior previously trained under controlled conditions. We confine ourselves principally to the problem of inferring the most likely pose given incomplete observations, where the missing data has already been labeled. In contrast to previous works that reconstruct face shapes from monocular views without a preexisting prior, we also solve for correspondence between the 2D tracked points and the prior model (and implicitly between views). We do not specifically address outlier rejection: In our experiments, the missing data was manually identified, but estimates of confidence (as in Torresani *et al.* [6], for example) could also be used to label missing data. We also do not address head orientation or the 2D tracking problem explicitly, but we believe our model can be extended to incorporate these as well.

In this paper we will demonstrate that a full covariance prior can be used successfully without a subspace model, provide a concise formulation of the MAP triangulation estimate for such a prior using multiple views, and describe a linear approximation for the perspective case that works well even for proximal cameras. However, our primary contribution is a Bayesian approach to formulating the correspondence problem, and an approximate solution to this formulation using linear assignment and local search that has been evaluated using facial motion capture data.

2 Related work

Recent work has emphasized low-rank factorization of facial motion into shape and pose. A seminal work by Blanz and Vetter [7] acquired models using 3D scanning, and used nonlinear optimization to fit to images. More recently, similar morphable models have been applied to monocular tracking [8] without a precomputed prior model. Torresani *et al.* [9] and Brand [10, 4, 3] present frameworks that work directly on video data, solving the tracking problem as part of the process. Del Bue *et al.* [11] have incorporated stereo views into a similar approach, but do not address the correspondence problem. In contrast to these approaches, we construct a prior in advance and use stereo views when available, but our approach handles monocular views and fully inferred points gracefully as well. In this sense our approach has more in common with active appearance models, which have been demonstrated to work well under occlusion [5], and also have been extended to stereo views [12] with full 3D models [13]. However, all of the aforementioned works use subspace models with a low number of modes. In Section 6 we will consider the optimum reconstructions possible with such models.

Bedekar and Haralick [14] have previously approached the problems of multiview triangulation and correspondence from a Bayesian perspective. Starink and Backer [15] address the correspondence problem using simulated annealing. Li *et al.* [16] and Chui and Rangajaran [17] specifically considered the case of non-rigid point correspondence. In contrast, our approach takes advantage of a Gaussian prior for the deformations of the underlying 3D model.

3 Prior acquisition

At the core of our approach, we assume that a person-specific prior can be constructed with high precision in a "calibration" process. This prior is then used to infer the correct shape from noisy, partially occluded 2D tracking data. We represent face shape at a fixed number n of 3D points, representing a given pose using the column vector $\mathbf{\Theta}$ with length $3n$. Using a set of calibrated 3D poses, we construct a full-covariance Gaussian prior:

$$P(\mathbf{\Theta}) = \mathcal{N}_{\mathbf{\Theta}}(\mu, \mathbf{\Sigma}) \tag{1}$$

We acquired our test and training data sets using a Vicon [2] optical motion capture system. Infra-red reflective markers were placed on an actor's face, and a number of performances were recorded using the system. Our test data consisted of dialog performed in a variety of emotional states. We used 1000 frames for our test set and 1964 frames for the training set. We stabilized the data by creating a coordinate system out of three points which move the least on the face. Specifically, we used the tops of the left and right ears, and a point on the bridge of the nose. Due to the stabilization of these points, the covariance matrix computed from this data is noninvertible. Even the stabilization points do still have some motion along certain axes, so rather than removing them from the training set we add isotropic noise ($\sigma = .01$ cm) along the diagonal to eliminate the zero singular values. The results given in Section 6 use a covariance matrix conditioned in this way.

Our 2D test data was created by projecting the 3D points through virtual orthographic and perspective cameras. In order to evaluate our approach under severe perspective distortion, our virtual perspective cameras used a wide angle lens (with a 127 degree angle of view) and were placed just 16cm from the center of the head.

4 Orthographic triangulation with a Gaussian prior

Let m_i represent the number of points visible in view i, and $\mathbf{p}_i$ the length-$2m_i$ vector of 2D points seen from view i. We represent the transformation of 3D model points to 2D view points as

$$\mathbf{p}_i = \mathbf{V}_i\mathbf{R}_i(\mathbf{M}_i\mathbf{\Theta} + \beta_i) + \mathbf{n}, \tag{2}$$

where the $3n \times 3n$ matrix $\mathbf{M}_i$ and $3n$-vector β represent the camera rotation and translation, $\mathbf{R}_i$ is a $3n \times 3n$ permutation matrix giving the correspondence of points in the model to points in view i (Section 5), $\mathbf{V}_i$ is a $2m \times 3n$ submatrix of the identity matrix that projects out the $n - m$ occluded points and the z-coordinate of the remaining points, and $\mathbf{n}$ is zero-mean Gaussian noise. In Section 5 we will solve for $\mathbf{R}_i$, but for simplicity, in this section we simplify this as

$$\mathbf{p}_i = \mathbf{N}_i\mathbf{\Theta} + \mathbf{b}_i + \mathbf{n}, \tag{3}$$

where $\mathbf{N}_i = \mathbf{V}_i\mathbf{R}_i\mathbf{M}_i$ is a $3n$ by $2m_i$ block-diagonal matrix and $\mathbf{b}_i = \mathbf{V}_i\mathbf{R}_i\beta_i$ is a length-$2m_i$ translation vector. Let σ represent the standard deviation of the pixel error, so that we characterize the probability model for our observations as:

$$P(\mathbf{p}_i|\mathbf{\Theta}) = \mathcal{N}_{p_i}(\mathbf{N}_i\mathbf{\Theta} + \mathbf{b}_i, \sigma^2\mathbf{I}) \tag{4}$$

Given some observed $\mathbf{p} = \{\mathbf{p}_1, \mathbf{p}_2, \cdots\}$, we wish to find the maximum a posteriori configuration of the model. Using Bayes' rule, the negative log likelihood is

$$-L(\mathbf{\Theta}|p) \sim -\sum_i L(\mathbf{p}_i|\mathbf{\Theta}) - L(\mathbf{\Theta}) \tag{5}$$

The log likelihoods are expanded as follows:

$$\begin{aligned} -L(\mathbf{\Theta}) &= \frac{1}{2}(\mathbf{\Theta} - \mu)^T\mathbf{\Sigma}^{-1}(\mathbf{\Theta} - \mu) + Z_{\mathbf{\Sigma}} & (6) \\ -L(\mathbf{p}_i|\mathbf{\Theta}) &= \|\mathbf{N}_i\mathbf{\Theta} + \mathbf{b}_i - \mathbf{p}_i\|^2/(2\sigma^2) + Z_\sigma & (7) \end{aligned}$$

(where $Z_{\mathbf{\Sigma}}$ and Z_σ are integration constants that do not depend on $\mathbf{\Theta}$ and can therefore be ignored in the formulation that follows.)

We wish to maximize the (log) likelihood of $\mathbf{\Theta}$. This is accomplished by differentiating by $\mathbf{\Theta}$ and setting to zero:

$$-\frac{dL(\mathbf{\Theta}|\mathbf{p}_i)}{d\mathbf{\Theta}} = \sum_i \mathbf{N}_i^T(\mathbf{N}_i\mathbf{\Theta} + \mathbf{b}_i - \mathbf{p}_i)/\sigma^2 + \mathbf{\Sigma}^{-1}(\mathbf{\Theta} - \mu) = 0 \tag{8}$$

$$\left(\sum_i \mathbf{N}_i^T\mathbf{N}_i + \sigma^2\mathbf{\Sigma}^{-1}\right)\mathbf{\Theta} = \sum_i \mathbf{N}_i^T(\mathbf{p}_i - \mathbf{b}_i) + \sigma^2\mathbf{\Sigma}^{-1}\mu \tag{9}$$

Equation 9 is linear in $\mathbf{\Theta}$ and can therefore be solved easily. Although in practice the visibility of points p_i (and therefore the matrices $\mathbf{N}_i$) may change from one frame to the next, if they change infrequently then many of the terms can be precomputed, including the inverse of the left hand side of Equation 9:

$$\begin{aligned} \mathbf{A} &= \left(\sum_i \mathbf{N}_i^T\mathbf{N}_i + \sigma^2\mathbf{\Sigma}^{-1}\right)^{-1} & (10) \\ \mathbf{B} &= \left(\mathbf{\Sigma}\sum_i \mathbf{N}_i^T\mathbf{N}_i/\sigma^2 + I\right)^{-1} & (11) \\ \mathbf{\Theta} &= \sum_i(\mathbf{A}\mathbf{N}_i^T)\mathbf{p}_i - \sum_i(\mathbf{A}\mathbf{N}_i^T)b_i + \mathbf{B}\mu & (12) \end{aligned}$$

The solution is just a single matrix multiply per view and one vector addition, since all the matrices $\mathbf{A}\mathbf{N}_i^T$ and the vector $-\sum_i(\mathbf{A}\mathbf{N}_i^T)b_i + \mathbf{B}\mu$ can be precomputed.

As in previous works applying factorization [8], $\mathbf{\Theta}$ can be replaced by $\mathbf{S}\theta$, where $\mathbf{S}$ is a reduced-dimensionality shape basis, and if the Gaussian model is not known a priori, $\mathbf{\Sigma}$ and μ can be updated online in order to progressively improve the model over time. Reducing the dimensionality with PCA reduces the size of many of the above matrices, resulting in improved performance at the cost of some accuracy. However, in Section 6 we will argue that the number of required bases is much larger than suggested by previous authors.

4.1 Extension to perspective cameras

Although the algorithm described in the previous section is an exact maximum a posteriori estimator for orthographic cameras, it does not hold for perspective cameras. However, this approach can be used in an iterative optimization of the full nonlinear function. In this setting $\mathbf{N}_i$ is linearized using the first-order Taylor expansion of the perspective transform at $\mathbf{\Theta}$. $\mathbf{\Theta}$ is initialized to either μ (on the first frame of a performance capture) or the pose on the previous frame $\mathbf{\Theta}_{t-1}$. The perspective transform is linearized each iteration, and the value of $\mathbf{\Theta}$ is updated using Equation 9. As long as points do not cross the eye plane, the system converges quickly, to machine precision in 1-3 iterations when initializing using the previous frame's pose, and only 6-7 iterations even when initializing to μ on each frame.

In order to linearize the perspective transform, we note that a perspective camera takes 4-d homogeneous points and projects them into 2-d, typically written as a 4×4 matrix $\mathbf{C}$. We can decompose $\mathbf{C}$ as follows:

$$\mathbf{C} = \begin{pmatrix} \bar{\mathbf{C}} & \mathbf{t} \\ \mathbf{C}_3^T & b \\ \mathbf{h}^T & c \end{pmatrix} \tag{13}$$

where $\bar{\mathbf{C}}$ is the upper-left 2×3 submatrix of $\mathbf{C}$, $\mathbf{t}$ is a column 2-vector, $\mathbf{C}_3$ and $\mathbf{h}$ are column 3-vectors, and b and c are scalars. Then, we rewrite the standard homogeneous transformation as

$$\mathbf{p}_2 = (\bar{\mathbf{C}}\mathbf{p}_3 + \mathbf{t})(\mathbf{h}^T\mathbf{p}_3 + c)^{-1} \tag{14}$$

(Note that $\mathbf{C}_3$ and b vanish, because we discard the z-coordinate when projecting to 2-d.)

Differentiating and massaging to standard form we find the Jacobian:

$$\frac{d\mathbf{p}_2}{d\mathbf{p}_3} = \frac{\bar{\mathbf{C}}}{\mathbf{h}^T\mathbf{p}_3 + c} - \frac{(\bar{\mathbf{C}}\mathbf{p}_3 + \mathbf{t})\mathbf{h}^T}{(\mathbf{h}^T\mathbf{p}_3 + c)^2} \tag{15}$$

Notice that, as expected, when $\mathbf{h} = \mathbf{0}$ and $c = 1$, the transformation is orthographic, and the Jacobian is equal to the 2×3 submatrix $\bar{\mathbf{C}}$. This approximation is stable as long as $\mathbf{h}^T\mathbf{p}_3 + c$ does not approach zero.

5 A Bayesian approach for correspondence

The preceding analysis presumes that the correspondences are known *a priori*, but the prior shape model is also useful in estimating the correspondences as well.

In this section we return to the complete projection model for view i given by Equation 2, in which the correspondence from 3D model points to points in 2D view i is given by $\mathbf{R}_i$. We wish to solve for the most likely permutation by maximizing $P(\mathbf{R}_i|\mathbf{p}_i)$. This expression is marginalized over all possible models $\mathbf{\Theta}$:

$$P(\mathbf{R}_i|\mathbf{p}_i) \sim P(\mathbf{p}_i|\mathbf{R}_i) = \int_{\mathbb{R}^{3n}} P(\mathbf{p}_i|\mathbf{R}_i, \mathbf{\Theta})P(\mathbf{\Theta})d\mathbf{\Theta} \tag{16}$$

We can compute this integral in closed form as follows:

$$\begin{aligned} P(\mathbf{p}_i|\mathbf{R}_i) &= \int_{\mathbb{R}^{3n}} P(\mathbf{p}_i|\mathbf{R}_i,\boldsymbol{\Theta})P(\boldsymbol{\Theta})d\boldsymbol{\Theta} && (17)\\ &= \int_{\mathbb{R}^{3n}} \mathcal{N}_{\mathbf{V}_i\mathbf{R}_i(\mathbf{M}_i\boldsymbol{\Theta}+\beta_i)}(\mathbf{p}_i,\sigma^2\mathbf{I})\mathcal{N}_{\boldsymbol{\Theta}}(\mu,\boldsymbol{\Sigma})d\boldsymbol{\Theta} && (18) \end{aligned}$$

The two Gaussians vary over different random variables, so we have to use a change of variables to align to the camera of view i: Define $\boldsymbol{\Omega} = \mathbf{R}_i\mathbf{M}_i\boldsymbol{\Theta}$. Furthermore, $\boldsymbol{\Omega}$ is partitioned into the variables that are "observed" (with noise) $\boldsymbol{\Omega}_o = \mathbf{V}_i\boldsymbol{\Omega}$ and the remainder of "unobserved" variables $\boldsymbol{\Omega}_u$ that are truly hidden. Now we can proceed:

$$P(\mathbf{p}_i|\mathbf{R}_i) = \int_{\mathbb{R}^{3n}} \mathcal{N}_{\boldsymbol{\Omega}_o+\mathbf{V}_i\mathbf{R}_i\beta_i}(\mathbf{p}_i,\sigma^2\mathbf{I})\mathcal{N}_{\boldsymbol{\Omega}}(\mathbf{R}_i\mathbf{M}_i\mu,\mathbf{R}_i\mathbf{M}_i\boldsymbol{\Sigma}(\mathbf{R}_i\mathbf{M}_i)^T)d\boldsymbol{\Omega} \quad (19)$$

Since the first of these Gaussians is independent of $\boldsymbol{\Omega}_u$, we integrate first over those variables, then over the rest:

$$\int\limits_{\mathbb{R}^{2m}} \mathcal{N}_{\boldsymbol{\Omega}_o}(\mathbf{p}_i - \mathbf{V}_i\mathbf{R}_i\beta_i,\sigma^2\mathbf{I}) \left(\int\limits_{\mathbb{R}^{3n-2m}} \mathcal{N}_{\boldsymbol{\Omega}}(\mathbf{R}_i\mathbf{M}_i\mu,\mathbf{R}_i\mathbf{M}_i\boldsymbol{\Sigma}(\mathbf{R}_i\mathbf{M}_i)^T)d\boldsymbol{\Omega}_u \right) d\boldsymbol{\Omega}_o \quad (20)$$

The inner integral is the marginal distribution over $\boldsymbol{\Omega}_u$, where the mean is simply the first $2m_i$ entries of $\mathbf{R}_i\mathbf{M}_i\mu$, and the covariance is the upper $2m_i \times 2m_i$ submatrix of $\mathbf{R}_i\mathbf{M}_i\boldsymbol{\Sigma}(\mathbf{R}_i\mathbf{M}_i)^T$. We apply $\mathbf{V}_i$ to concisely represent these as:

$$\begin{aligned} \mu_o^* &= \mathbf{V}_i\mathbf{R}_i\mathbf{M}_i\mu && (21)\\ \boldsymbol{\Sigma}_o^* &= \mathbf{V}_i\mathbf{R}_i\mathbf{M}_i\boldsymbol{\Sigma}(\mathbf{V}_i\mathbf{R}_i\mathbf{M}_i)^T && (22)\\ P(\mathbf{p}_i|\mathbf{R}_i) &= \int_{\mathbb{R}^{2m}} \mathcal{N}_{\boldsymbol{\Omega}_o}(\mathbf{p}_i - \mathbf{V}_i\mathbf{R}_i\beta_i,\sigma^2\mathbf{I})\mathcal{N}_{\boldsymbol{\Omega}_o}(\mu_o^*,\boldsymbol{\Sigma}_o^*)d\boldsymbol{\Omega}_o && (23) \end{aligned}$$

The product of Gaussians can now be simplified [18] as

$$\begin{aligned} P(\mathbf{p}_i|\mathbf{R}_i) &= \int_{\mathbb{R}^{2m}} \mathcal{N}_{\mathbf{p}_i-\mathbf{V}_i\mathbf{R}_i\beta_i}(\mu_o^*,\sigma^2\mathbf{I}+\boldsymbol{\Sigma}_o^*)\mathcal{N}_{\boldsymbol{\Omega}_o}(\mu_c,\boldsymbol{\Sigma}_c)d\boldsymbol{\Omega}_o && (24)\\ &= \mathcal{N}_{\mathbf{p}_i-\mathbf{V}_i\mathbf{R}_i\beta_i}(\mu_o^*,\sigma^2\mathbf{I}+\boldsymbol{\Sigma}_o^*)\int_{\mathbb{R}^{2m}} \mathcal{N}_{\boldsymbol{\Omega}_o}(\mu_c,\boldsymbol{\Sigma}_c)d\boldsymbol{\Omega}_o && (25)\\ &= \mathcal{N}_{\mathbf{p}_i}(\mu_o^* + \mathbf{V}_i\mathbf{R}_i\beta_i,\sigma^2\mathbf{I}+\boldsymbol{\Sigma}_o^*) && (26)\\ &= \mathcal{N}_{\mathbf{p}_i}(\mathbf{V}_i\mathbf{R}_i(\mathbf{M}_i\mu+\beta_i),\mathbf{V}_i\mathbf{R}_i(\sigma^2\mathbf{I}+\mathbf{M}_i\boldsymbol{\Sigma}\mathbf{M}_i^T)(\mathbf{V}_i\mathbf{R}_i)^T) && (27) \end{aligned}$$

(The new variables μ_c and $\boldsymbol{\Sigma}_c$ introduced in Equation 24 can also be expressed in terms of the means and covariances of the original Gaussians, but as that distribution is marginalized out in the subsequent equations, we omit the full formula here for brevity.)

Multiple frames of data can be used to estimate correspondence, by taking the product of Equation 27 over all frames: $P(\mathbf{p}_i|\mathbf{R}_i) = \prod_t P(\mathbf{p}_i(t)|\mathbf{R}_i)$.

In special cases, the log of Equation 27 can be shown equivalent to the quadratic assignment problem (see Appendix A), which is known to be NP-hard [19]. The complete problem is therefore challenging to solve optimally. However, we have utilized the following approach, and found that it works well in practice: First we approximate the solution using linear assignment (i.e. assuming independence between points), which often produces the correct result and otherwise is usually a short distance away from the optimal solution. Then we approach the optimum using a short local search, by swapping pairs of point assignments that increase the likelihood given by Equation 27. Although not guaranteed to find the optimal solution, we have found that these initialization and local search approaches often reach the optimum in less than a dozen swaps (see Section 6).

Note that this approach solves for correspondences from a view to the model, rather than explicitly between views. Therefore it is performed independently per view, and correspondences between views can be recovered implicitly. [1]

5.1 Initialization via linear assignment

The linear assignment problem can be expressed [20] as the minimization of a product of weights and costs $\sum_{j,k} \mathbf{C}(j,k)\mathbf{X}(j,k)$, subject to the constraint that the weights $\mathbf{X}$ are a binary permutation matrix. The optimum can be found in polynomial time using linear programming by relaxing the constraints on $\mathbf{X}$ to

$$\sum_k \mathbf{X}(j,k) = 1 \qquad \forall j \tag{28}$$

$$\sum_j \mathbf{X}(j,k) = 1 \qquad \forall k \tag{29}$$

$$\mathbf{X}(j,k) \geq 0 \qquad \forall j,k \tag{30}$$

To convert Equation 27 to a form suitable for linear assignment, we take the negative log likelihood of a single assignment (i.e. a single column of $\mathbf{R}_i$) marginalized over all other point positions. This leads to the following formula for entries of the cost matrix:

$$\mathbf{C}(j,k) = (\mathbf{p}_{i(j)} - \mu^*_{(k)} - \beta_{i(k)})^T(\sigma^2\mathbf{I} + \mathbf{\Sigma}^*_{(k)})^{-1}(\mathbf{p}_{i(j)} - \mu^*_{(k)} - \beta_{i(k)}) \tag{31}$$

where $p_{i(j)}$ is the jth 2D point in p_i, $\mu^*_{(k)}$ is the kth 2D mean in $N_i\mu$, $\beta_{i(k)}$ is the kth 2D translation in β_i, and $\Sigma^*_{(k)}$ is the kth 2×2 block along the diagonal of $N_i \Sigma N_i^T$. Note that we can omit the log normalization constant from these costs, since their sum contributes a constant amount to the final cost. For similar reasons, entries of the matrix corresponding to missing data ($j > m_i$) are set to 0, since there is neither a cost nor benefit to these assignments. Again, in the case where multiple frames are used to estimate correspondence, we simply sum up the cost matrices for each frame.

[1] Since we have marginalized out the model, we can no longer linearize a perspective projection about $\mathbf{\Theta}$ as in Section 4.1. Therefore when solving for correspondence for perspective cameras we have used the Taylor series expansion at μ instead.

5.2 Optimization via pairwise swaps

Starting from the initial permutation obtained via linear assignment, we modify R_i via all $n(n-1)$ pairwise swaps of rows and columns, and test the resulting cost function in Equation 27 to find the best pairwise swap. This process is iterated until convergence, ie. no better configuration can be found.

The costliest computation in the process is the inversion of the covariance matrix $\mathbf{V}_i\mathbf{R}_i(\sigma^2\mathbf{I}+\mathbf{M}_i\mathbf{\Sigma}\mathbf{M}_i^T)(\mathbf{V}_i\mathbf{R}_i)^T$, which changes for each pairwise swap in $\mathbf{R}_i$. However, we can accelerate the method considerably by classifying the swaps into three types:

1. Swapping two occluded points $j_0 > m_i$ and $j_1 > m_i$;
2. Swapping two observed points $j_0 \leq m_i$ and $j_1 \leq m_i$; and
3. Swapping an observed point $j_0 \leq m_i$ with an occluded point $j_1 > m_i$.

The first type of swap does not alter the matrix or indeed the entire cost function, and we can therefore omit computations for all $(n-m)(n-m-1)$ of these swaps entirely.

The second type of swap does alter the cost function, but as it merely permutes rows and columns in the covariance matrix, the inverse is also such a permutation. To see this, let $\mathbf{S}$ represent such a swap, and note that $\mathbf{S}$ is orthonormal, so $\mathbf{S}^T = \mathbf{S}^{-1}$, so $(\mathbf{S}\mathbf{A}\mathbf{S}^T)^{-1} = \mathbf{S}\mathbf{A}^{-1}\mathbf{S}^T$ for any invertible $\mathbf{A}$. Also note that the determinant is unchanged by the swap, so the normalization constant need not be recomputed.

The third type is the only one that actually requires nontrivial computation, but when $n-m$ is small we use Schur complements to compute it efficiently. The Schur complement relates the inverse of a large matrix to the inverses of two smaller matrices, and is ordinarily used when inverting large matrices. Here however, we apply it less conventionally, inverting a matrix using the precomputed inverse of another large matrix and the inverse of a smaller matrix that varies.

Specifically, we precompute the $2n \times 2n$ matrix

$$\mathbf{Q}^{-1} = (\mathbf{V}_{xy}(\sigma^2\mathbf{I}+\mathbf{M}_i\mathbf{\Sigma}\mathbf{M}_i^T)\mathbf{V}_{xy}^T)^{-1} \tag{32}$$

where $\mathbf{V}_{xy}$ is defined as the $2n\times 3n$ submatrix of $\mathbf{I}$ that extracts the x and y coordinates of each point (i.e. $\mathbf{V}$ for a view in which no points are occluded). Now construct the permutation $\mathbf{R}_\mathbf{Q}$ such that the upper left block of $\mathbf{R}_\mathbf{Q}\mathbf{Q}\mathbf{R}_\mathbf{Q}^T$ is the matrix to be inverted, $\mathbf{V}_i\mathbf{R}_i(\sigma^2\mathbf{I}+\mathbf{M}_i\Sigma\mathbf{M}_i^T)(\mathbf{V}_i\mathbf{R}_i)^T$. Then compute:

$$(\mathbf{R}_\mathbf{Q}\mathbf{Q}\mathbf{R}_\mathbf{Q}^T)^{-1} = \mathbf{R}_\mathbf{Q}\mathbf{Q}^{-1}\mathbf{R}_\mathbf{Q}^T = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{D} \end{pmatrix}, \tag{33}$$

partitioned into blocks $\mathbf{A} : 2m \times 2m$, $\mathbf{B} : 2m \times 2(n-m)$, and $\mathbf{D} : 2(n-m) \times 2(n-m)$. Finally, take the Schur complement of $\mathbf{D}$ in $(\mathbf{R}_\mathbf{Q}\mathbf{Q}\mathbf{R}_\mathbf{Q}^T)^{-1}$ to obtain the desired inverse:

$$(\mathbf{V}_i\mathbf{R}_i(\sigma^2\mathbf{I}+\mathbf{M}_i\mathbf{\Sigma}\mathbf{M}_i^T)(\mathbf{V}_i\mathbf{R}_i)^T)^{-1} = \mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{B}^T \tag{34}$$

In short, for each pairwise swap of the third type, we only need to invert a matrix of size $2(n-m_i) \times 2(n-m_i)$; all other computations are matrix multiplications or permutations.

Table 1: RMS and max errors (in cm) for optimal reconstruction of face shapes tracked at 127 points using different numbers of PCA bases. Note that even for 10 modes, the maximum error for a point can be as much as .3cm. When reprojected at film or HD resolution, such an error represents a deviation of up to 30 pixels.

$K =$	**4** [4]	**6** [3]	**10** [8]	**21**	**40**	**100**	**300**
RMS error	0.066	0.063	0.045	0.029	0.015	0.005	0.001
max error	0.191	0.202	0.296	0.085	0.056	0.015	0.003

6 Results

In this section we present an analysis of PCA projection for face shapes with different numbers of modes, and results for our triangulation and correspondence approaches.

6.1 Optimal reconstruction using PCA bases

Much recent work in face tracking has relied on low-dimensional subspace models to constrain the solution space of poses. This restriction is quite powerful, as it makes it possible to learn pose variation even without training data. However, in this paper we argue that such models are not sufficient for high-quality facial motion capture, since even optimal reconstructions using such a model have significant error. In particular, the traditional use of SVD to determine the number of modes uses an L_2 error norm. We believe a better metric is to minimize the L_∞ error norm, to constrain the worst-case deviation of a reconstruction from ground truth. To assess the performance of different numbers of bases we applied PCA to our training data set and tested RMS and max errors for different numbers of PCA bases, summarized in Table 1.

As the number of modes increases, online learning of the model and pose becomes more challenging (especially with unknown camera pose). Hence we use the full-dimensional Gaussian, but we do not attempt to learn the prior model or camera pose online, relying instead on high-quality training data to construct our prior.

6.2 Triangulation

To evaluate our triangulation approach, we applied it against our test data in four settings, using 1 or 2 virtual cameras with either orthographic and perspective projections, with $\sigma = 1 \times 10^{-5}$. For each setting, we increased the number of occluded points from 0% to 25%, changing randomly each frame. The results of these tests are shown in Figure 1. RMS error is below 1mm in all cases. Max error is considerably better for multiple views, but even with 15% of points occluded in a single perspective view, the maximum error over all 1000 frames of test data is less than 0.5cm.

6.3 Correspondence

To evaluate our approach for solving the correspondence problem, we ran our algorithm 200 times on different single frames of test data in 4 settings: orthographic and

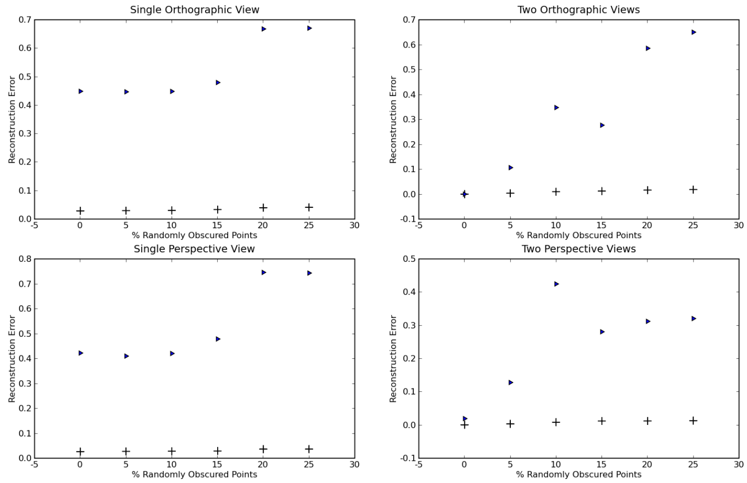


Figure 1: RMS (black crosses) and maximum error (blue triangles) rates (in cm) for our triangulation approach, graphed vs. number of occluded points for one orthographic camera (top left), one perspective camera (bottom right), two orthographic cameras (top right), and two perspective cameras (bottom right).

perspective cameras, with all points visible and with 15% points occluded. Linear assignment took approximately 3s per frame, while pairwise swaps took anywhere from 90s to 140s per frame to converge, using a 3.0GHz Intel Core Duo CPU. The results of these tests are shown in Figures 2 and 3.

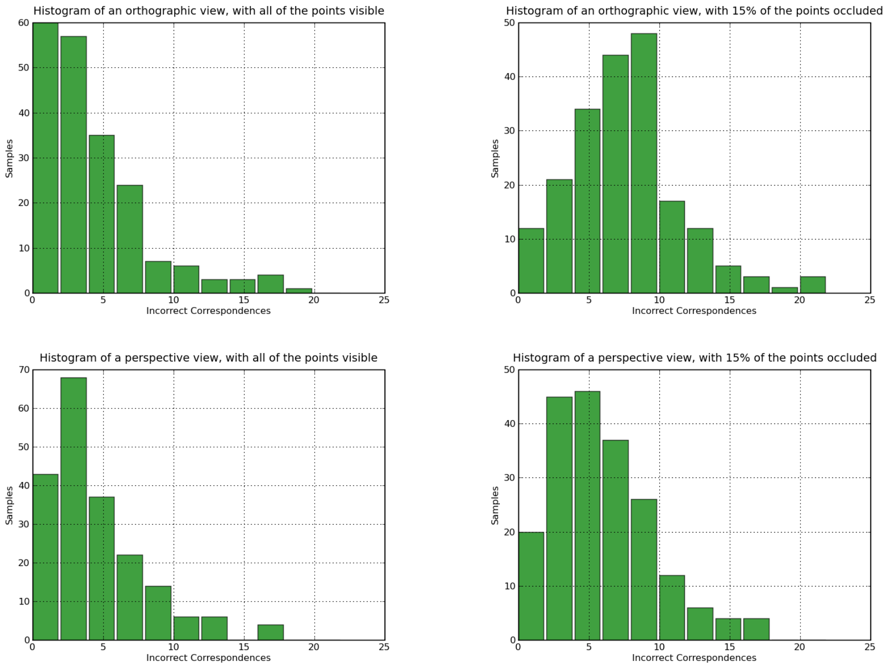
In all four settings, linear assignment works fairly well for initialization, as the majority of frames have 10 or fewer incorrect correspondences. Unsurprisingly, performance is better when all points are visible.

The pairwise swap local search performs extremely well when all points are visible, but poorly in the occluded case. In fact, for perspective cameras with occlusions there are often more incorrect correspondences after pairwise swaps than before. We plan to investigate this case more fully in future work.

We have not yet investigated the use of multiple frames for estimating a single correspondence, but we expect error rates to drop considerably when multiple frames are available for estimation.

7 Conclusion

In this work we have presented the use of a full-covariance Gaussian prior to aid in triangulation and correspondence in the presence of missing data. We have demonstrated that such a prior can be used successfully without a subspace model, and the resulting MAP triangulation estimate is simple to compute for either orthographic or perspective



Histogram of an orthographic view, with all of the points visible

Histogram of an orthographic view, with 15% of the points occluded

Histogram of a perspective view, with all of the points visible

Histogram of a perspective view, with 15% of the points occluded

Figure 2: Histograms of number of incorrect correspondences after initialization with linear assignment, for an orthographic view with all points visible (top left), an orthographic view with 15% of the points randomly occluded on each frame (top right), a perspective view with 0% of points randomly occluded on each frame (bottom left), and a perspective view with 15% of points randomly occluded on each frame. All histograms are unnormalized, with 200 sample frames.

cameras. Furthermore we believe our analysis and formulation of the correspondence problem is a particularly useful contribution, and our approximate solution is both fast and accurate.

In the future we hope to expand the functionality of our approach, for example by exploring the use of a full-covariance Gaussian prior to estimate both camera pose and model pose simultaneously, applying online learning techniques, and integrating tracking from video directly into our formulation, to aid in robust occlusion-handling.

References

[1] Zhang, L., Snavely, N., Curless, B., Seitz, S.M.: Spacetime faces: High-resolution capture for modeling and animation. ACM Trans. on Graphics (Proc. SIGGRAPH) **23** (2004) 548–558

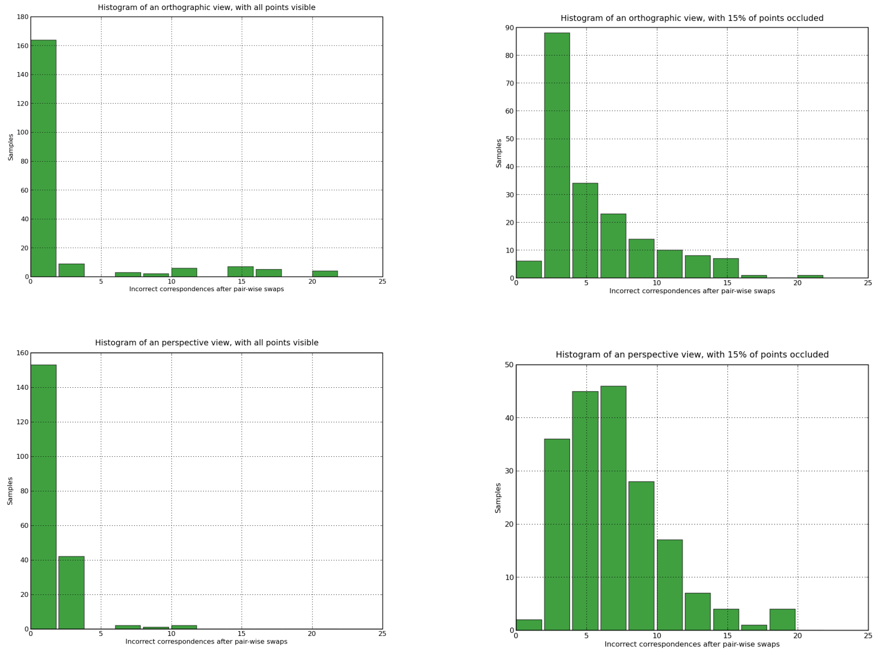


Figure 3: Histograms of number of incorrect correspondences after pairwise swap optimization, for an orthographic view with all points visible (top left), an orthographic view with 15% of the points randomly occluded on each frame (top right), a perspective view with 0% of points randomly occluded on each frame (bottom left), and a perspective view with 15% of points randomly occluded on each frame. All histograms are unnormalized, with 200 sample frames.

[2] Vicon: Motion capture systems from Vicon. [Accessed online at http://www.vicon.com/] (2008)

[3] Brand, M.: A direct method for 3d factorization of nonrigid motion observed in 2d. In: Proc. CVPR. Volume 2. (2005) 122–128

[4] Brand, M.: Morphable 3D models from video. In: Proc. CVPR. (2001)

[5] Gross, R., Matthews, I., Baker, S.: Active appearance models with occlusion. Image and Vision Computing **24** (2006) 593–604

[6] Torresani, L., Hertzmann, A., Bregler, C.: Robust model-free tracking of nonrigid shape. Technical Report TR2003-840, New York University (2003)

[7] Blanz, V., Vetter, T.: A morphable model for the synthesis of 3d faces. In: Proc. SIGGRAPH '99. (1999) 187–194

[8] Torresani, L., Hertzmann, A., Bregler, C.: Non-rigid structure-from-motion: Estimating shape and motion with hierarchical priors. PAMI (2008) (to appear)

[9] Torresani, L., Hertzmann, A.: Automatic non-rigid 3d modeling from video. In: Proc. ECCV. Volume II. (2004) 299–312

[10] Brand, M., Bhotika, R.: Flexible flow for 3D nonrigid tracking and shape recovery. In: CVPR. Volume 1. (2001) 315–322

[11] Bue, A.D., Agapito, L.: Non-rigid stereo factorization. IJCV **66** (2006) 193–207

[12] Hu, C., Xiao, J., Matthews, I., Baker, S., Cohn, J., Kanade, T.: Fitting a single active appearance model simultaneously to multiple images. In: Proc. of the British Machine Vision Conference. (2004)

[13] Xiao, J., Baker, S., Matthews, I., Kanade, T.: Real-time combined 2d+3d active appearance models. In: Proc. CVPR. Volume 2. (2004) 535–542

[14] Bedekar, A.S., Haralick, R.M.: Finding corresponding points based on bayesian triangulation. cvpr **00** (1996) 61

[15] Starink, J., Backer, E.: Finding point correspondences using simulated annealing. **28** (1995) 231–240

[16] Li, B., Meng, Q., Holstein, H.: Similarity k-d tree method for sparse point pattern matching with underlying non-rigidity. **38** (2005) 2391–2399

[17] Chui, H., Rangarajan, A.: A new point matching algorithm for non-rigid registration. Comput. Vis. Image Underst. **89** (2003) 114–141

[18] Petersen, K.B., Pedersen, M.S.: The matrix cookbook. [Accessed online at http://matrixcookbook.com/] (2008)

[19] Pardalos, P., Rendl, F., Wolkowicz, H.: The quadratic assignment problem: a survey and recent developments. In Pardalos, P., Wolkowicz, H., eds.: Quadratic assignment and related problems (New Brunswick, NJ, 1993). Amer. Math. Soc., Providence, RI (1994) 1–42

[20] Wikipedia: Assignment problem — wikipedia, the free encyclopedia (2008) [Online; accessed 16-March-2008].

A Quadratic assignment problem

The quadratic assignment problem can be defined as follows [19]:

$$\min_{\mathbf{X}\in\mathbf{\Pi}} f(\mathbf{X}) = \mathrm{Tr}\left[(\mathbf{AXB}+\mathbf{C})\mathbf{X}^T\right] \tag{35}$$

where $\mathbf{\Pi}$ is the space of permutation matrices, and Tr is the trace operator.

Consider the special case of Equation 27 in which the model consists of 2D points, viewed in 1-dimensional cameras. Thus μ is a $2n$-vector and Σ is a $2n \times 2n$ symmetric matrix, while $\mathbf{p}$ is an n-vector. Note that due to the structure of $\mathbf{V}$ and $\mathbf{R}$, the odd elements of $\mathbf{M}_i\mu + \beta_i$ and odd rows and columns of $\sigma^2\mathbf{I} + \mathbf{M}_i\boldsymbol{\Sigma}\mathbf{M}_i^T$ will always be projected out, so we can write the equivalent expression

$$P(\mathbf{p}|\mathbf{R}) = \mathcal{N}_{\mathbf{p}}(\mathbf{Rx}, \mathbf{RWR}) \tag{36}$$

where $\mathbf{R}$ is now a general $n \times n$ permutation matrix, $\mathbf{x}$ is a constant length n-vector, $\mathbf{W}$ is a constant symmetric $n \times n$ matrix. (We omit subscripts i for clarity.)

Taking the negative log, and letting α_1 and α_2 represent constant scale and offset factors we find

$$\begin{aligned} -\alpha_1 L(\mathbf{p}|\mathbf{R}) + \alpha_2 &= (\mathbf{p} - \mathbf{Rx})^T(\mathbf{RWR}^T)^{-1}(\mathbf{p} - \mathbf{Rx}) && (37)\\ &= \text{Tr}\left[(\mathbf{p} - \mathbf{Rx})(\mathbf{p} - \mathbf{Rx})^T\mathbf{RW}^{-1}\mathbf{R}^T\right] && (38)\\ &= \text{Tr}\left[(\mathbf{pp}^T - 2\mathbf{Rxp}^T + \mathbf{Rxx}^T\mathbf{R}^T)\mathbf{RW}^{-1}\mathbf{R}^T\right] && (39)\\ &= \text{Tr}\left[\mathbf{pp}^T\mathbf{RW}^{-1}\mathbf{R}^T\right] \\ &\quad -2\text{Tr}\left[\mathbf{Rxp}^T\mathbf{RW}^{-1}\mathbf{R}^T\right] + \text{Tr}\left[\mathbf{Rxx}^T\mathbf{W}^{-1}\mathbf{R}^T\right] && (40)\end{aligned}$$

Using the property $\text{Tr}\left[\mathbf{AB}\right] = \text{Tr}\left[\mathbf{BA}\right]$, simplify to

$$-\alpha_1 L(\mathbf{p}|\mathbf{R}) + \alpha_2 = \text{Tr}\left[\mathbf{pp}^T\mathbf{RW}^{-1}\mathbf{R}^T\right] - 2\text{Tr}\left[\mathbf{W}^{-1}\mathbf{xp}^T\mathbf{R}\right] + \text{Tr}\left[\mathbf{xx}^T\mathbf{W}^{-1}\right] \tag{41}$$

The third term is constant w.r.t. $\mathbf{R}$ and can be folded into α_2, and using the property $\text{Tr}\left[\mathbf{A}^T\right] = \text{Tr}\left[\mathbf{A}\right]$, this simplifies again to

$$\begin{aligned} -\alpha_1 L(\mathbf{p}|\mathbf{R}) + \alpha_2 &= \text{Tr}\left[\mathbf{pp}^T\mathbf{RW}^{-1}\mathbf{R}^T\right] - 2\text{Tr}\left[\mathbf{px}^T\mathbf{W}^{-1}\mathbf{R}^T\right] && (42)\\ &= \text{Tr}\left[\mathbf{pp}^T\mathbf{RW}^{-1}\mathbf{R}^T - 2\mathbf{px}^T\mathbf{W}^{-1}\mathbf{R}^T\right] && (43)\\ &= \text{Tr}\left[(\mathbf{pp}^T\mathbf{RW}^{-1} - 2\mathbf{px}^T\mathbf{W}^{-1})\mathbf{R}^T\right] && (44)\end{aligned}$$

Thus we have a QAP of the form in Equation 35 such that

$$\begin{aligned} \mathbf{A} &= \mathbf{pp}^T && (45)\\ \mathbf{B} &= \mathbf{W}^{-1} && (46)\\ \mathbf{C} &= -2\mathbf{px}^T\mathbf{W}^{-1}. && (47)\end{aligned}$$

We conjecture that the full correspondence problem described in Section 5 is therefore also in the same complexity equivalence class, but have no formal proof at this time.